Государственное образовательное учреждение

высшего профессионального образования

«Уфимский Государственный Авиационный Технический Университет»

Кафедра «Вычислительной математики и кибернетики»

Отчет

По лабораторной работе

по дисциплине: Компьютерная графика

Лабораторная работа № 2. Пространственный чертеж с изображением  
отрезка и плоскости заданной треугольником.

Проверил

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Выполнили:

студенты гр. ПРО-301в

Доронин С.Г.

Хабутдинов Р.А.

Уфа 2013

# Задание:

Разработать программу, позволяющую изображать отрезок, плоскость заданную треугольникам и их проекции на пространственном и комплексном чертежах с возможностью изменять координаты точек отрезка и треугольника.

# Содержание экрана:

* Пространственный чертеж (первый октант) с изображением отрезка MN и плоскости (заданной треугольником АВС) с изображением точки их пересечения Т;
* комплексный чертеж (6ез профильной проекции.) с изображением проекций отрезка, плоскости, точки пересечения и линий связи (для всех 6 точек);
* ползунковые переключатели (3 шт.) для интерактивного изменения координат (x,y,z) точек М, N, А, В, С (с возможностью выбора одной из 5 исходных точек для которой в текущий момент времени будет осуществляться изменение координат).

# Динамика:

При изменении координат 5 исходных точек должны изменяться соответствующие проекции и сами исходные ГО, а также точка их пересечения. Невидимая по отношению к наблюдателю, находящемуся в бесконечности на положительной полуоси Z, часть отрезка должна быть выделена. Плоскость, заданная треугольником ABC, считается бесконечной и непрозрачной.

\* - для определения коэффициентов плоскости используется метод Ньюэла,

\*\* - исходные графические объекты могут находиться только в первом октанте.

# Структура решения

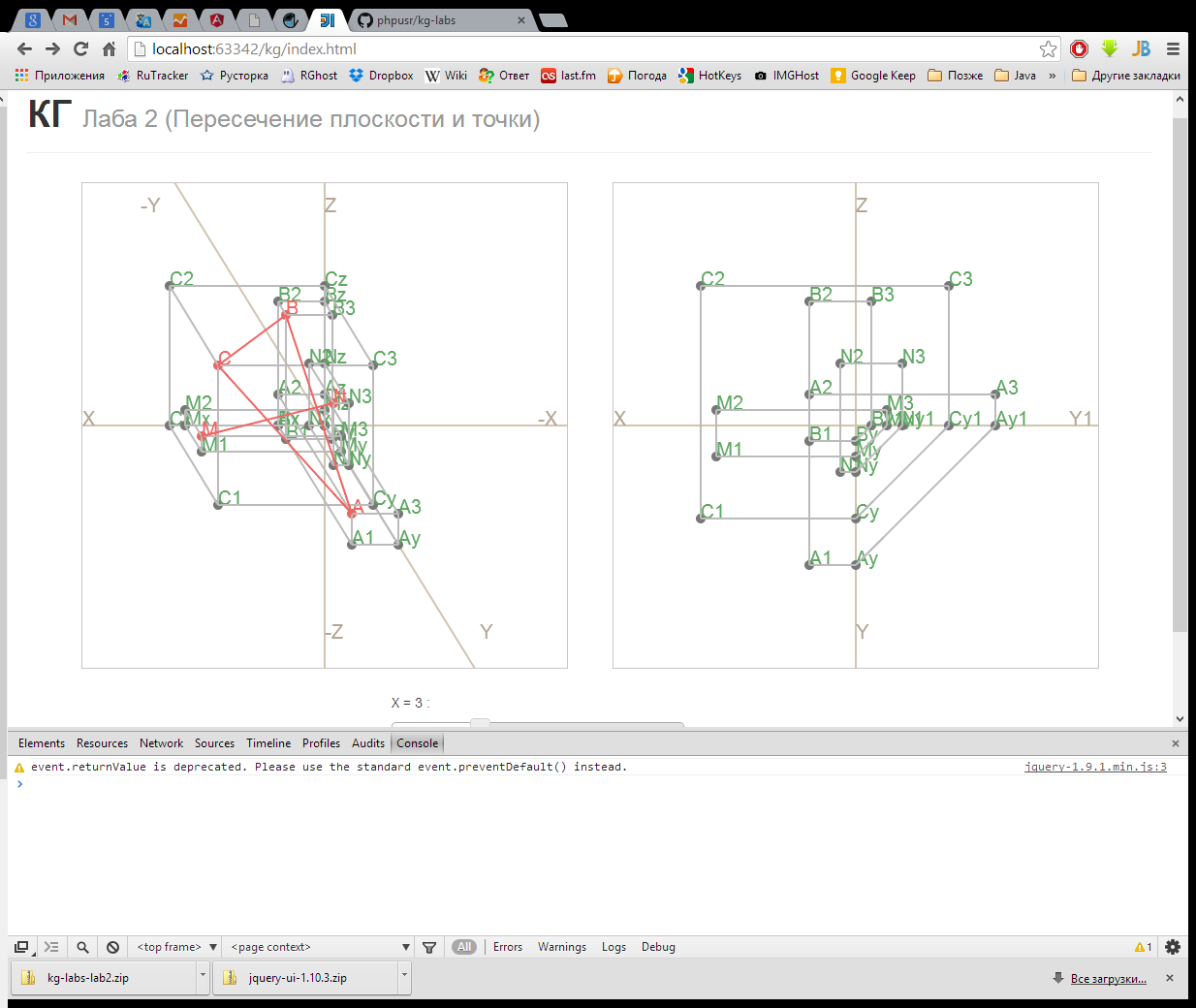
При изменении пользователем координат (x, y, z) точки с помощью ползунковых переключателей в программу передаются управляющие параметры, исходя из которых, изменяются чертежи. Схема изменения выглядит так:

1. Вычисление коэффициентов плоскости;
2. Нахождение точки пересечения плоскости и прямой;
3. Определение видимости отрезка;
4. Пересчет реальных координат в координаты экрана:
   * для пространственного чертежа;
   * для комплексного чертежа;
5. Отрисовка плоскости и прямой:
   * для пространственного чертежа;
   * для комплексного чертежа.

# Обзор и анализ методов решения

Пусть имеется точки с координатам A(XA,YA,ZA), B(XB,YB,ZB), C(XC,YC,ZC), M(XM, YM, ZM), N(XN,YN,ZN).

## Построение пространственного чертежа



### Вычисление коэффициентов плоскости

Уравнение плоскости: ax + by + cz + d = 0.

Через три не коллинеарные точки. Если подставить координаты этих точек в уравнение плоскости, пронормировать относительно коэффициента D, то получим систему трех уравнений в котором имеются три неизвестные A, B и C. Решив эту систему мы получим соответствующие коэффициенты, а подставив их и взяв координаты произвольной точки, лежащей на плоскости мы получим коэффициент d. Поскольку объем вычисления, во многих алгоритмах компьютерной графики растет с увеличением числа многоугольников и для описания плоскостей, часть используют многоугольники с более чем тремя сторонами, а эти многоугольники могут быть не только выпуклыми, но и самое главное не плоскими, то для подсчета коэффициентов плоскостей для этого случая известен метод Ньюэла, который решает эту задачу и для неплоских многоугольников. Этот метод эквивалентен определению нормали в каждой вершине многоугольника посредством векторного произведения прилежащих ребер и усреднения результатов.

Метод Ньюэла.

(вычисление коэффициентов (a, b, c, d) плоскости, заданной n вершинами,

в общем случае, не лежащих на одной плоскости)

Построение проекции точек по заданным координатам осуществляется в следующей последовательности:

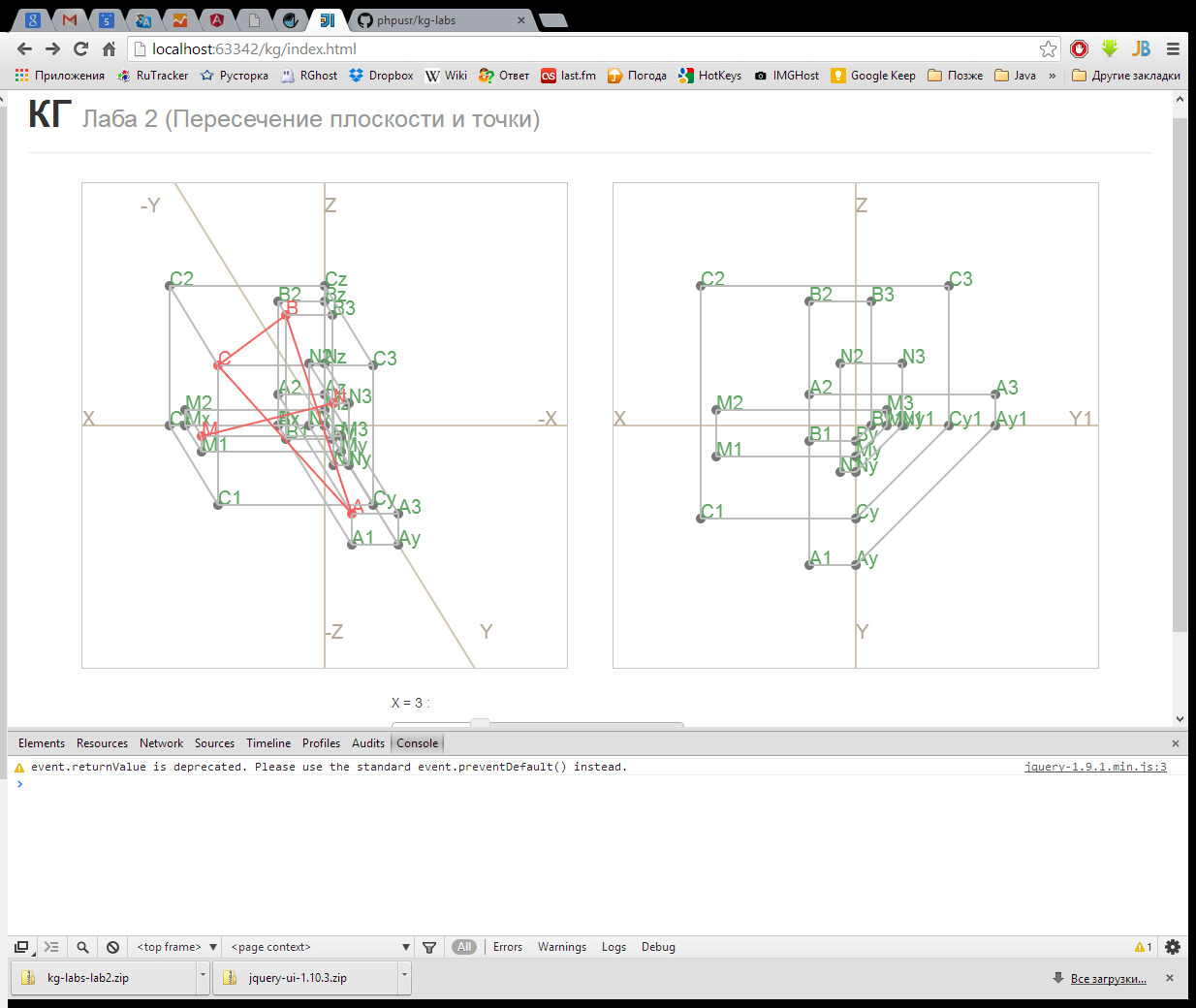
* на оси абсцисс от начала координат откладывают отрезок ОАх=х
* через точку Аx проводят перпендикуляр к оси Ох, на котором с учетом знаков откладывают отрезки АхА1=у (получаем A1-горизонтальную проекцию) AxA2=z (получаем А2-фронтальную проекцию)
* через А2 проводят перпендикуляр к оси Oz
* с учетом знака откладывают отрезок АzА3=у (получаем А3-профильную проекцию)

### Определение видимости отрезка

За направление плоскости возьмем такое, при котором скалярное произведение ее на точку наблюдения дает положительный результат. Если после определения коэффициентов плоскости скалярное их произведение на точку наблюдения (0100), расположенную в бесконечности на положительной полуоси Y дает "-", то необходимо умножить коэффициенты плоскости на "-1".

Такое направление нормали плоскости позволяет использовать следующее правило для определения видимости концов отрезка:если скалярное произведение конца отрезка на столбец коэффициентов плоскости положительно, то эта вершина видима.

## 2. Построение комплексного чертежа



На координатной плоскости XZ изобразим пространственное изображение. Ось Y образовывает угол с осью X 45 градусов. Точка A(x, y, z) на координатной плоскости XZ будет иметь координаты (x1, y1), где

x1=x-y\*cos(45);

y1=z-y\*sin(45);

Горизонтальная проекция имеет координаты A1 (x1, y1-z).

Фронтальная проекция имеет координаты A2 (х1+y\*cos(45), y1 + y\*cos(45)). Профильная проекция имеет координаты A3 (x1-x, y1).

Построение параллелепипеда осуществляется следующим образом:

-соединить точку А с полученными точками A1, A2, Тз.

-из A1 провести прямую к оси ОХ параллельную OX параллельную к оси OY

-из A2 опустить перпендикуляры на оси OX, OZ

-из A3 провести прямую к оси OZ параллельную OY и к оси OY параллельную OZ

# Описание реализации применяемых методов

Переменные, которые используются в реализации: x3D, y3D, z3D - координаты точки, вводимые пользователем

х, у - координаты точки на пространственном.

X0, Y0 - величина является началом координат в пространственном и комплексном чертежах (X0, Y0) (B координатах устройства).

## 1. Преобразование координат точки в координаты устройства для построение пространственного чертежа.

Система координат устройства имеет вид:

X

Y

Исходя из этого координаты точки в координатах устройства будут вычисляться таким образом:

x = X0 - Math.round((x3D - y3D \* Math.cos(ALPHA)) \* SCALE);

y = Y0 - Math.round((z3D - y3D \* Math.sin(ALPHA)) \* SCALE);

А проекции будут иметь координаты:

A1x= x-y\*cosα

A1y= -y\*sinα

A2x=-y\*cosα

A2y= z-y\*sinα

A3x= x

A3y= z

## 2. Преобразование координат точки в координаты устройства для построение комплексного чертежа

Аналогично предыдущему пункту, вследствие того что устройство имеет свою систему координат, нужно преобразовать координаты. Проекции будут иметь координаты:

A1(X0-A1x, Y0-A1y);

A2(X0-A2x, Y0-A2y);

A3(X0-A3x, Y0-A3y);

# Описание программного обеспечения

Программа на Javascript:

'use strict'**;**

/\*\*

\* Класс: Точка

\*/

***function*** Point**(**drawing**,** x**,** y**,** x3D**,** y3D**,** z3D**)** **{**

***this*.**x **=** x**;**

***this*.**y **=** y**;**

***this*.**x3D **=** x3D**;**

***this*.**y3D **=** y3D**;**

***this*.**z3D **=** z3D**;**

/\*\* Рисование точки \*/

***this*.**drawPoint **=** ***function*(**pointName**,** color**)** **{**

drawing**.**drawCircle**(*this*.**x**,** ***this*.**y**,** 5**,** color**);**

***if*** **(**pointName**)** ***this*.**drawText**(**pointName**,** color**);**

***return*** ***this*;**

**};**

/\*\* Рисование текста \*/

***this*.**drawText **=** ***function*** **(**text**,** color**)** **{**

drawing**.**drawTextXY**(**text**,** ***this*.**x**,** ***this*.**y**,** color**);**

***return*** ***this*;**

**};**

/\*\* Рисование линии \*/

***this*.**drawLine **=** ***function*** **(**pointTo**,** color**)** **{**

drawing**.**drawLineXY**(*this*.**x**,** ***this*.**y**,** pointTo**.**x**,** pointTo**.**y**,** color**);**

***return*** ***this*;**

**};**

/\*\* Рисование проекции точки \*/

***this*.**drawProjection **=** ***function*(**text**)** **{**

//Проекция точки A

***var*** point0 **=** drawing**.**createPoint3D**(**0**,** 0**,** 0**).**drawPoint**();**

***var*** pointAx **=** drawing**.**createPoint3D**(*this*.**x3D**,** 0**,** 0**).**drawPoint**(**text **+** 'x'**);**

***var*** pointAy **=** drawing**.**createPoint3D**(**0**,** ***this*.**y3D**,** 0**).**drawPoint**(**text **+** 'y'**);**

***var*** pointAz **=** drawing**.**createPoint3D**(**0**,** 0**,** ***this*.**z3D**).**drawPoint**(**text **+** 'z'**);**

***var*** pointA1 **=** drawing**.**createPoint3D**(*this*.**x3D**,** ***this*.**y3D**,** 0**).**drawPoint**(**text **+** '1'**);**

***var*** pointA2 **=** drawing**.**createPoint3D**(*this*.**x3D**,** 0**,** ***this*.**z3D**).**drawPoint**(**text **+** '2'**);**

***var*** pointA3 **=** drawing**.**createPoint3D**(**0**,** ***this*.**y3D**,** ***this*.**z3D**).**drawPoint**(**text **+** '3'**);**

point0**.**drawLine**(**pointAy**).**drawLine**(**pointAz**).**drawLine**(**pointAx**);**

pointAz**.**drawLine**(**pointA3**).**drawLine**(**pointA2**);**

pointAy**.**drawLine**(**pointA3**).**drawLine**(**pointA1**);**

pointAx**.**drawLine**(**pointA2**).**drawLine**(**pointA1**);**

//Точка A

***var*** pointA **=** drawing**.**createPoint3D**(*this*.**x3D**,** ***this*.**y3D**,** ***this*.**z3D**).**drawPoint**(**text**,** COLOR\_HIGHLIGHT**);**

pointA**.**drawLine**(**pointA1**).**drawLine**(**pointA2**).**drawLine**(**pointA3**);**

**};**

/\*\* Рисование комплексного чертежа точки \*/

***this*.**drawComplex **=** ***function*(**drawing**,** text**)** **{**

***var*** pointA1 **=** drawing**.**createPoint2D**(*this*.**x3D**,** **-*this*.**y3D**).**drawPoint**(**text **+** '1'**);**

***var*** pointA2 **=** drawing**.**createPoint2D**(*this*.**x3D**,** ***this*.**z3D**).**drawPoint**(**text **+** '2'**);**

***var*** pointA3 **=** drawing**.**createPoint2D**(-*this*.**y3D**,** ***this*.**z3D**).**drawPoint**(**text **+** '3'**);**

***var*** pointAy **=** drawing**.**createPoint2D**(**0**,** **-*this*.**y3D**).**drawPoint**(**text **+** 'y'**);**

***var*** pointAy1 **=** drawing**.**createPoint2D**(-*this*.**y3D**,** 0**).**drawPoint**(**text **+** 'y1'**);**

pointAy**.**drawLine**(**pointAy1**).**drawLine**(**pointA1**);**

pointA3**.**drawLine**(**pointAy1**).**drawLine**(**pointA2**);**

pointA1**.**drawLine**(**pointA2**);**

**};**

**}**

/\*\*

\* Класс: Drawing

\*/

***function*** Drawing**(**canvas**,** type**)** **{**

***var*** ALPHA **=** 45**;**

***var*** \_self **=** ***this*;**

/\*\* Создание 3D точки \*/

***this*.**createPoint3D **=** ***function*(**x3D**,** y3D**,** z3D**)** **{**

***var*** x **=** getX**(**x3D**,** y3D**);**

***var*** y **=** getY**(**z3D**,** y3D**);**

***return*** ***new*** Point**(*this*,** x**,** y**,** x3D**,** y3D**,** z3D**);**

**};**

/\*\* Создание 2D точки \*/

***this*.**createPoint2D **=** ***function*(**x2D**,** y2D**)** **{**

***var*** x **=** getX**(**x2D**,** 0**);**

***var*** y **=** getY**(**y2D**,** 0**);**

***return*** ***new*** Point**(*this*,** x**,** y**);**

**};**

/\*\* X - координата в 2D \*/

***function*** getX**(**x3D**,** y3D**)** **{**

***var*** v1 **=** X0 **-** **(**x3D **-** Math**.**round**(**y3D **\*** Math**.**cos**(**ALPHA**)))** **\*** SCALE**;** //Криво по Y

***var*** v2 **=** X0 **-** Math**.**round**((**x3D **-** y3D **\*** Math**.**cos**(**ALPHA**))** **\*** SCALE**);** //Нормально по Y

***return*** v2**;**

**}**

/\*\* Y - координата в 2D \*/

***function*** getY**(**z3D**,** y3D**)** **{**

***var*** v1 **=** Y0 **-** **(**z3D **-** Math**.**round**(**y3D **\*** Math**.**sin**(**ALPHA**)))** **\*** SCALE**;**

***var*** v2 **=** Y0 **-** Math**.**round**((**z3D **-** y3D **\*** Math**.**sin**(**ALPHA**))** **\*** SCALE**);**

***return*** v2**;**

**}**

/\*\* Рисование текста \*/

***this*.**drawTextXY **=** ***function*(**text**,** x**,** y**,** color**)** **{**

canvas**.**fillStyle **=** color **?** color **:** COLOR\_TEXT**;**

canvas**.**font **=** 'normal 15pt Arial'**;**

canvas**.**fillText**(**text**,** x**,** y**);**

**};**

/\*\* Рисование линии \*/

***this*.**drawLineXY **=** ***function*(**x1**,** y1**,** x2**,** y2**,** color**)** **{**

canvas**.**beginPath**();**

canvas**.**moveTo**(**x1**,** y1**);**

canvas**.**lineTo**(**x2**,** y2**);**

canvas**.**lineWidth **=** 2**;**

canvas**.**strokeStyle **=** color **?** color **:** COLOR\_LINE**;**

canvas**.**stroke**();**

**};**

/\*\* Рисование окружности \*/

***this*.**drawCircle **=** ***function*(**x**,** y**,** radius**,** color**)** **{**

canvas**.**fillStyle **=** color **?** color **:** COLOR\_POINT**;**

canvas**.**beginPath**();**

canvas**.**arc**(**x**,** y**,** radius**,** 0**,** Math**.**PI**\***2**,** ***false*);**

canvas**.**closePath**();**

canvas**.**fill**();**

**};**

/\*\* Очистка канвы \*/

***this*.**clearCanvas **=** ***function*()** **{**

canvas**.**clearRect**(**MIN\_X**,** MIN\_Y**,** MAX\_X**,** MAX\_Y**);**

**};**

/\*\* Построение осей координат \*/

***this*.**drawAxis **=** ***function*()** **{**

***this*.**clearCanvas**();**

//Рисование осей

***if*** **(**type **==** 'dim'**)** **{**

***var*** y1 **=** ***this*.**createPoint3D**(**0**,** **-**100**,** 0**);**

***var*** y2 **=** ***this*.**createPoint3D**(**0**,** 100**,** 0**);**

y1**.**drawLine**(**y2**,** COLOR\_AXIS**);**

**}**

***this*.**drawLineXY**(**X0**,** MIN\_Y**,** X0**,** MAX\_Y**,** COLOR\_AXIS**);**

***this*.**drawLineXY**(**MIN\_X**,** Y0**,** MAX\_X**,** Y0**,** COLOR\_AXIS**);**

//Заголовки осей

***var*** size **=** 30**;**

***var*** titles **=** **{**

dim**:** **[{**title**:** 'X'**,** x**:** MIN\_X**,** y**:** Y0**},** **{**title**:** '-X'**,** x**:** MAX\_X**-**size**,** y**:** Y0**},**

**{**title**:** 'Z'**,** x**:** X0**,** y**:** MIN\_Y**+**size**},** **{**title**:** '-Z'**,** x**:** X0**,** y**:** MAX\_Y**-**size**},**

**{**title**:** 'Y'**,** x**:** MAX\_X**-**size**\***3**,** y**:** MAX\_Y**-**size**},** **{**title**:** '-Y'**,** x**:** MIN\_X**+**size**\***2**,** y**:** MIN\_Y**+**size**}],**

cmplx**:** **[{**title**:** 'X'**,** x**:** MIN\_X**,** y**:** Y0**},** **{**title**:** 'Y1'**,** x**:** MAX\_X**-**size**,** y**:** Y0**},**

**{**title**:** 'Z'**,** x**:** X0**,** y**:** MIN\_Y**+**size**},** **{**title**:** 'Y'**,** x**:** X0**,** y**:** MAX\_Y**-**size**}]**

**};**

$**.**each**(**titles**[**type**],** ***function*(**index**,** value**)** **{**

\_self**.**drawTextXY**(**value**.**title**,** value**.**x**,** value**.**y**,** COLOR\_AXIS\_TITLE**);**

**});**

**};**

**}**

'use strict'**;**

/\*\* App Module \*/

***var*** app **=** angular**.**module**(**'myApp'**,** **[**'ngSanitize'**]);**

app**.**controller**(**'MyCtrl'**,** ***function*** MyCtrl**(**$scope**)** **{**

$scope**.**subject **=** 'КГ'**;**

$scope**.**lab **=** 'Лаба 2 (Пересечение плоскости и точки)'**;**

$scope**.**notSupport **=** 'Браузер не поддерживает Canvas'**;**

//Канва

***var*** cnvDim **=** document**.**getElementById**(**'canvasDimensional'**).**getContext**(**"2d"**);**

***var*** cnvCmplx **=** document**.**getElementById**(**'canvasComplex'**).**getContext**(**"2d"**);**

$**(**'#canvasDimensional'**).**attr**(**'width'**,** MAX\_X**).**attr**(**'height'**,** MAX\_Y**);**

$**(**'#canvasComplex'**).**attr**(**'width'**,** MAX\_X**).**attr**(**'height'**,** MAX\_Y**);**

$**(**'#drawBlock'**).**css**(**'width'**,** BLOCK\_WIDTH**);**

$**(**'#slider-block'**).**css**(**'width'**,** SLIDER\_WIDTH**);**

***var*** drwDim **=** ***new*** Drawing**(**cnvDim**,** 'dim'**);**

***var*** drwCmplx **=** ***new*** Drawing**(**cnvCmplx**,** 'cmplx'**);**

***var*** points **=** **[**

**{**text**:** 'A'**,** p**:** drwDim**.**createPoint3D**(**3**,**7**,**2**)},**

**{**text**:** 'B'**,** p**:** drwDim**.**createPoint3D**(**3**,**1**,**8**)},**

**{**text**:** 'C'**,** p**:** drwDim**.**createPoint3D**(**10**,**6**,**9**)},**

**{**text**:** 'M'**,** p**:** drwDim**.**createPoint3D**(**9**,**2**,**1**)},**

**{**text**:** 'N'**,** p**:** drwDim**.**createPoint3D**(**1**,**3**,**4**)}**

**];**

$scope**.**points **=** points**;**

***var*** current **=** **{**point**:** 0**};**

$scope**.**current **=** current**;**

/\*\* Инициализация слайдеров \*/

***function*** initSliders**()** **{**

$**(**'#sliderX'**).**slider**({**

range**:** 'max'**,** min**:** 0**,** max**:** 10**,** value**:** 0**,**

slide**:** ***function*(**event**,** ui**)** **{**

repaintDrawing**(**ui**.**value**,** null**,** null**);**

**}**

**});**

$**(**'#sliderY'**).**slider**({**

range**:** 'max'**,** min**:** 0**,** max**:** 10**,** value**:** 0**,**

slide**:** ***function*(**event**,** ui**)** **{**

repaintDrawing**(**null**,** ui**.**value**,** null**);**

**}**

**});**

$**(**'#sliderZ'**).**slider**({**

range**:** 'max'**,** min**:** 0**,** max**:** 10**,** value**:** 0**,**

slide**:** ***function*(**event**,** ui**)** **{**

repaintDrawing**(**null**,** null**,** ui**.**value**);**

**}**

**});**

**}**

$scope**.**changePoint **=** ***function*()** **{**

***var*** point **=** points**[**current**.**point**].**p**;**

$**(**'#sliderX'**).**slider**(**'value'**,** point**.**x3D**);**

$**(**'#sliderY'**).**slider**(**'value'**,** point**.**y3D**);**

$**(**'#sliderZ'**).**slider**(**'value'**,** point**.**z3D**);**

repaintDrawing**();**

**};**

/\*\* Перерисовка чертежей \*/

***function*** repaintDrawing**(**valX**,** valY **,**valZ**)** **{**

valX **=** valX **!=** null **?** valX **:** $**(**'#sliderX'**).**slider**(**'value'**);**

valY **=** valY **!=** null **?** valY **:** $**(**'#sliderY'**).**slider**(**'value'**);**

valZ **=** valZ **!=** null **?** valZ **:** $**(**'#sliderZ'**).**slider**(**'value'**);**

//Отображение значений ползунков

$scope**.**valX **=** valX**;**

$scope**.**valY **=** valY**;**

$scope**.**valZ **=** valZ**;**

angularApply**(**$scope**);**

//Построение Пространственного чертежа

drawDimensional**(**valX**,** valY**,** valZ**);**

//Построение Комплексного чертежа

drawComplex**();**

**}**

***var*** pointT**;** //Точка T

/\*\* Построение Пространственного чертежа \*/

***function*** drawDimensional**(**valX**,** valY **,**valZ**)** **{**

//Очистка канвы и построение осей

drwDim**.**drawAxis**();**

points**[**current**.**point**].**p **=** drwDim**.**createPoint3D**(**valX**,** valY**,** valZ**);**

***var*** pointA **=** points**[**0**].**p**;**

***var*** pointB **=** points**[**1**].**p**;**

***var*** pointC **=** points**[**2**].**p**;**

***var*** pointM **=** points**[**3**].**p**;**

***var*** pointN **=** points**[**4**].**p**;**

pointT **=** null**;**

//Вычисление коэффициентов плоскости

***var*** A **=** **(**pointA**.**y3D**-**pointB**.**y3D**)** **\*** **(**pointA**.**z3D**+**pointB**.**z3D**)** **+** **(**pointB**.**y3D**-**pointC**.**y3D**)** **\*** **(**pointB**.**z3D**+**pointC**.**z3D**)** **+** **(**pointC**.**y3D**-**pointA**.**y3D**)** **\*** **(**pointC**.**z3D**+**pointA**.**z3D**);**

***var*** B **=** **(**pointA**.**z3D**-**pointB**.**z3D**)** **\*** **(**pointA**.**x3D**+**pointB**.**x3D**)** **+** **(**pointB**.**z3D**-**pointC**.**z3D**)** **\*** **(**pointB**.**x3D**+**pointC**.**x3D**)** **+** **(**pointC**.**z3D**-**pointA**.**z3D**)** **\*** **(**pointC**.**x3D**+**pointA**.**x3D**);**

***var*** C **=** **(**pointA**.**x3D**-**pointB**.**x3D**)** **\*** **(**pointA**.**y3D**+**pointB**.**y3D**)** **+** **(**pointB**.**x3D**-**pointC**.**x3D**)** **\*** **(**pointB**.**y3D**+**pointC**.**y3D**)** **+** **(**pointC**.**x3D**-**pointA**.**x3D**)** **\*** **(**pointC**.**y3D**+**pointA**.**y3D**);**

***var*** D **=** **-(**A **\*** pointA**.**x3D **+** B **\*** pointA**.**y3D **+** C **\*** pointA**.**z3D**);**

//Нахождение точки пересечения плоскости с прямой

***if*** **((**A**\*(**pointM**.**x3D **-** pointN**.**x3D**)** **+** B**\*(**pointM**.**y3D **-** pointN**.**y3D**)** **+** C**\*(**pointM**.**z3D **-** pointN**.**z3D**))** **!=** 0**)** **{**

***var*** t **=** **(**A**\***pointM**.**x3D **+** B**\***pointM**.**y3D **+** C**\***pointM**.**z3D **+** D**)** **/** **(**A**\*(**pointM**.**x3D**-**pointN**.**x3D**)** **+** B**\*(**pointM**.**y3D**-**pointN**.**y3D**)** **+** C**\*(**pointM**.**z3D**-**pointN**.**z3D**));**

***if*** **(**t**>=**0 **&&** t**<=**1**)** **{**

***var*** xT **=** pointM**.**x3D **+** **(**pointN**.**x3D**-**pointM**.**x3D**)\***t**;**

***var*** yT **=** pointM**.**y3D **+** **(**pointN**.**y3D**-**pointM**.**y3D**)\***t**;**

***var*** zT **=** pointM**.**z3D **+** **(**pointN**.**z3D**-**pointM**.**z3D**)\***t**;**

pointT **=** drwDim**.**createPoint3D**(**xT**,** yT**,** zT**);**

pointT**.**drawPoint**(**'T'**,** COLOR\_HIGHLIGHT**);**

**}**

**}**

//Определение видимости отрезка

***if*** **(**B**<**0**)** **{**

A **=** **-**A**;** B **=** **-**B**;** C **=** **-**C**;** D **=** **-**D**;**

**}**

***var*** visibleM**;**

***var*** tmpM **=** A**\***pointM**.**x3D **+** B**\***pointM**.**y3D **+** C**\***pointM**.**z3D **+** D**;**

***if*** **(**tmpM **>** 0**)** **{**

visibleM **=** ***true*;**

**}** ***else*** ***if*** **(**tmpM **<** 0**)** **{**

visibleM **=** ***false*;**

**}**

***var*** visibleN**;**

***var*** tmpN **=** A**\***pointN**.**x3D **+** B**\***pointN**.**y3D **+** C**\***pointN**.**z3D **+** D**;**

***var*** delta **=** 0.01**;** //TODO какое значение поставить?

***var*** tmpMN **=** **(**A**==**0 **&&** B**==**0 **&&** C**==**0**)** **||** **(**A**\*(**pointM**.**x3D**-**pointN**.**x3D**)** **+** B**\*(**pointM**.**y3D**-**pointN**.**y3D**)** **+** C**\*(**pointM**.**z3D**-**pointN**.**z3D**))==**0 **||** **(**Math**.**abs**(**B**)** **<** delta**);**

***if*** **(**tmpN **>** 0**)** **{**

visibleN **=** ***true*;**

**}** ***else*** ***if*** **(**tmpN **<** 0**)** **{**

visibleN **=** ***false*;**

**}** ***else*** ***if*** **(**tmpMN**)** **{**

visibleM **=** ***true*;**

visibleN **=** ***true*;**

**}**

//Построение чертежа

***for*** **(*var*** i**=** 0**;** i**<**points**.**length**;** i**++)** **{**

points**[**i**].**p**.**drawProjection**(**points**[**i**].**text**)**

**}**

pointA**.**drawLine**(**pointB**,** COLOR\_HIGHLIGHT**).**drawLine**(**pointC**,** COLOR\_HIGHLIGHT**);**

pointB**.**drawLine**(**pointC**,** COLOR\_HIGHLIGHT**);**

***if*** **(**visibleM**)** pointM**.**drawPoint**(**'M'**);**

***if*** **(**visibleN**)** pointN**.**drawPoint**(**'N'**);**

pointM**.**drawLine**(**pointN**,** COLOR\_HIGHLIGHT**);**

**}**

/\*\* Построение Комплексного чертежа \*/

***function*** drawComplex**()** **{**

//Очистка канвы и построение осей

drwCmplx**.**drawAxis**();**

***for*** **(*var*** i**=** 0**;** i**<**points**.**length**;** i**++)** **{**

points**[**i**].**p**.**drawComplex**(**drwCmplx**,** points**[**i**].**text**);**

**}**

***if*** **(**pointT**)** pointT**.**drawComplex**(**drwCmplx**,** 'T'**);**

**}**

//Start

initSliders**();**

$scope**.**changePoint**();**

**});**

# Пример выполнения программы

